Modelling Concurrent Systems Specified by Dynamic Information Systems: A Rough Set Approach

Krzysztof Pancerz [1] Zbigniew Suraj [2,3]

Chair of Computer Science Foundations
University of Information Technology and Management
Rzeszów, Poland

Abstract

This paper presents a new approach for modelling concurrent systems specified by the dynamic information systems. As a model for concurrency Coloured Petri Nets are used. The dynamic information systems contain knowledge about the dynamics of modelled concurrent systems.

Keywords: Behaviour, dynamic information systems, rough sets, concurrency, Coloured Petri Nets.

1 Introduction

A modelling of concurrent systems specified by information systems has recently been discussed in [4], [5], [7], [8], [9], [10], [12]. A description of concurrent systems by classical information systems does not respect their behaviour. In this paper we propose an approach to modelling of concurrent systems specified by dynamic information systems introduced in [11]. The behaviour of concurrent systems can easily be specified by dynamic information systems. We apply the Coloured Petri Nets (CP-nets) for modelling of concurrent systems specified by dynamic information systems. Models in the form of CP-nets are coherent, readable and their construction is simple. Informally, a dynamic information system DS can be defined as follows. Let $A = \{a_1, \ldots, a_m\}$ be a nonempty, finite set of processes. With every process $a \in A$ we associate a finite set V_a of its local states. The behaviour of a concurrent system is presented in the form of two integrated subtables denoted

[1] Email: kpancerz@wenus.wsiz.rzeszow.pl
[2] Email: zsuraj@wenus.wsiz.rzeszow.pl
[3] Institute of Mathematics of Rzeszow University, Poland

by S and TS. S represents global states of a given concurrent system and it is called an underlying system of DS. However TS represents the transition relation T between global states in the given system and it is called a transition system. Each row in the first subtable includes a record of local states of processes from A. Each record is labeled by an element from the set U of global states of the system. The second subtable represents a transition system. Columns of the second subtable are labeled by events, whilst rows, analogously as for the underlying system S, by objects of interest. Entries of the subtable for a given row (a global state) are follower states of that state. The first row of the underlying system S represents the initial state of a given transition system TS.

The problem is: For a given dynamic information system DS construct its concurrent model in the form of a Coloured Petri Net CPN_{DS} such that:
(1) The occurrence graph $OG(CPN_{DS})$ defines an extension DS' of DS created by adding to S all new global states corresponding to markings of CPN_{DS} and adding to TS all new transitions between new global states corresponding to arcs of $OG(CPN_{DS})$.
(2) All new global states in S' are consistent with all rules true in S as well as all new transitions between global states in TS' are consistent with rules true in TS.
(3) DS' is the largest extension of DS with those properties.

The methods for constructing concurrent models of dynamic information systems presented in this paper consist of two stages. In the first stage, all dependencies represented by rules are extracted from a given dynamic information system. We consider two sets of extracted rules. The first set of rules corresponds to dependencies between values of local states of processes in the system S. However the second set of rules corresponds to the transitions between global states of the system S. In the second stage, the nets corresponding to these dependencies are built. We assume that a given data table DS consists of only partial knowledge about the system behaviour. Constructed concurrent models in the form of a *CP*-net allow us to find an extension DS' of DS. In order to do it we construct two models. The first model is built for the underlying system S of DS by using the method described in [5], whereas the second model is built for the system DS by using new methods proposed in this paper. The first model allows us to find an extension S' of S. On the base of this extension and the second model we can determine an extension T' of the transition relation T from DS. The extension DS' includes all global states consistent with all rules true in the underlying information system S of DS. However, the transition system TS' of DS' corresponds to all possible transitions between global states from S' consistent with all rules generated for the transition system TS.

2 Basic definitions and notations

In this section we recall some notions and notations related to dynamic information systems [6], [11], [12] as well as Coloured Petri Nets [2].

An *information system* is a pair $S = (U, A)$, where U is a nonempty, finite set of *objects*, called the *universe*, A is a nonempty, finite set of *attributes*, i.e. $a : U \to V_a$ for $a \in A$, where V_a is called the *value set* of a. The set $V = \bigcup_{a \in A} V_a$ is said to be the domain of A. Any information system $S = (U, A)$ determines an *information function* $Inf_A : U \to 2^{A \times V}$ defined by $Inf_A(u) = \{(a, a(u)) : a \in A\}$. The values of an information function will be represented by vectors of the form $(v_1, \ldots, v_m)$, $v_i \in V_{a_i}$, $i = 1, \ldots, m$, where $m = card(A)$. Such vectors are called *information vectors* (over A and V).

A *decision table* is any information system of the form $S = (U, A \cup D)$, where $A \cap D = \emptyset$ and D is a set of distinguished attributes called *decisions*. The elements of A are called *conditional attributes* (*conditions*).

A *transition system* is a tuple $TS = (U, E, T, u_0)$, where U is a nonempty set of *states*, E is a set of *events*, $T \subseteq U \times E \times U$ is a *transition relation*, u_0 is the *initial state*. If $(u, e, u') \in T$ then a transition system TS can go from u to u' as a result of the event e occurring at u.

A *dynamic information system* is a tuple $DS = (U, A, E, T, u_0)$ where:
(i) $S = (U, A)$ is an information system called the *underlying system* of DS,
(ii) $TS = (U, E, T, u_0)$ is a transition system. The dynamic information systems can be presented in the form of two integrated subtables. The first subtable represents the underlying system S, whereas the second one the transition system TS.

Let $S = (U, A)$ be an information system, where $A = \{a_1, \ldots, a_m\}$ and V is the domain of A. Pairs (a, v), where $a \in A$, $v \in V$ are called *descriptors* over A and V. Instead of (a, v) we write also $a = v$ or a_v.

A *rule* (or *positive rule*) [12] of an information system S is any expression of the form $a_{i_1} = v_{i_1} \wedge \ldots \wedge a_{i_r} = v_{i_r} \Rightarrow a_p = v_p$, where $a_p, a_{i_j} \in A$, $v_p \in V_{a_p}$, $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, \ldots, r$ and $\wedge$, $\Rightarrow$ denote conjunction and implication, respectively (the classical propositional operators). The set of all optimal rules (i.e. rules with a minimal number of descriptors on the left hand side) in the information system S is denoted by $OPT(S)$.

A *decision rule* of an decision system S is any expression of the form $a_{i_1} = v_{i_1} \wedge \ldots \wedge a_{i_r} = v_{i_r} \Rightarrow a_d = v_d$, where $a_{i_j} \in A$, $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, \ldots, r$ and $a_d \in D$, $v_d \in V_{a_d}$.

An *inhibitor rule* [5] of an information system S is any expression of the form $a_{i_1} = v_{i_1} \wedge \ldots \wedge a_{i_r} = v_{i_r} \Rightarrow \neg(a_p = v_p)$, where $a_p, a_{i_j} \in A$, $v_p \in V_{a_p}$, $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, \ldots, r$ and $\neg$ denotes negation (the classical propositional operator). The set of all inhibitor rules in the information system S corresponding to the set $OPT(S)$ is denoted by $INH(S)$.

A *Coloured Petri Net* (*CP*-net) [2] is a tuple $CPN = (\Sigma, P, T, A, N,$

$C, G, E, I)$ satisfying the requirements below: Σ is a nonempty, finite set of types which are called *colour sets*, P is a finite set of *places*, T is a finite set of *transitions*, A is a finite set of *arcs*, $N : A \to (P \times T) \cup (T \times P)$ is a *node function*, $C : P \to \Sigma$ is a *colour function*, G is a *guard function*, E is an *arc expression function*, I is an *initialization function*. Furtheran CP-net corresponding to a dynamic information system DS will be denoted by CPN_{DS}. By $M[b\rangle M'$ we denote fact that the marking M' is directly reachable from the marking M for a binding element b. By $OG(M_0)$ we denote the occurrence graph of a CP-net, where M_0 is its initial marking.

Example 2.1 Let us consider an exemplary dynamic information system $DS = (U, A, E, T, u_1)$ as in Table 1 such that: the set of global states $U = \{u_1, u_2, u_3, u_4\}$, the set of local processes $A = \{a, b\}$, the set of events (actions) $E = \{e_1, e_2, e_3, e_4\}$, the transition relation $T = \{(u_1, e_1, u_4), (u_2, e_2, u_3), (u_3, e_3, u_1), (u_4, e_4, u_2)\}$ and u_1 represents the initial state of the system. The local states of processes from A are defined as in Table 1.

Table 1
A dynamic information system DS

$U\backslash A$	a	b	$U\backslash E$	e_1	e_2	e_3	e_4
u_1	0	1		u_4			
u_2	1	0			u_3		
u_3	0	2				u_1	
u_4	2	0					u_2

3 Specifications of concurrent systems

This section presents two approaches to specifications of concurrent systems by means of dynamic information systems. Both specifications lead to different results as it will be shown later.

Let $DS = (U, A, E, T, u_0)$ be a dynamic information system.

The first approach (a weak specification). In this approach we construct a decision table $DT_1 = (U_1, A \cup A')$, where a set of conditions $A = \{a_1, ..., a_m\}$ and a set of decisions $A' = \{a'_1, ..., a'_m\}$. Such table contains some pairs of global states from the underlying system S of DS. Each row of the decision table corresponds to a transition between the global states u, u' determined by the transition relation T, i.e. for which there exists an event $e \in E$ such that $(u, e, u') \in T$. The conditional attributes $a_1, ..., a_m$ correspond to the previous global states u, whereas the decision attributes $a'_1, ..., a'_m$ correspond to the next global states u'.

Example 3.1 Let us consider a dynamic information system DS from Example 2.1. A decision table $DT_1 = (U_1, A \cup A')$ has the form as in Table 2.

Table 2
A decision table DT_1

$U_1 \backslash A \cup A'$	a	b	a'	b'	$U_1 \backslash A \cup A'$	a	b	a'	b'
u_1	0	1	2	0	u_3	0	2	0	1
u_2	1	0	0	2	u_4	2	0	1	0

The second approach (a strong specification). This approach has been proposed in [12]. We construct a decision table $DT_2 = (U_2, A_2 \cup \{d\})$, where $A_2 = A \cup A'$, $A = \{a_1, ..., a_m\}$ and $A' = \{a'_1, ..., a'_m\}$. Such table contains all possible pairs of global states from the underlying system S of DS. The attributes $a_1, ..., a_m$ and $a'_1, ..., a'_m$ are conditions and determine the previous global states u and the next global states u', respectively, whereas d is a decision. The value of decision d is equal to 1 iff there exists an event $e \in E$ such that $(u, e, u') \in T$ and 0 otherwise.

Example 3.2 Let us consider again a dynamic information system DS from Example 2.1. A decision table $DT_2 = (U_2, A_2 \cup \{d\})$ has the form as in Table 3.

Table 3
A decision table DT_2

$U_2 \backslash A_2$	a	b	a'	b'	d	$U_2 \backslash A_2$	a	b	a'	b'	d
u_1	0	1	0	1	0	u_9	0	2	0	1	1
u_2	0	1	1	0	0	u_{10}	0	2	1	0	0
u_3	0	1	0	2	0	u_{11}	0	2	0	2	0
u_4	0	1	2	0	1	u_{12}	0	2	2	0	0
u_5	1	0	0	1	0	u_{13}	2	0	0	1	0
u_6	1	0	1	0	0	u_{14}	2	0	1	0	1
u_7	1	0	0	2	1	u_{15}	2	0	0	2	0
u_8	1	0	2	0	0	u_{16}	2	0	2	0	0

Remark 3.3 It is worth to observe that the first approach is - in some sense - a particular case of the second one. The decision table DT_1 includes only those pairs of global states from the table DT_2 for which the decision attribute d has the value 1.

4 Modelling of concurrent systems

Let $DS = (U, A, E, T, u_0)$ be a dynamic information system. Below we describe two algorithms for constructing concurrent models in the form of CP-nets corresponding to a given dynamic information system DS on the base of specifications presented in section 3.

ALGORITHM 1 for constructing a concurrent model CPN_{DS1} of DS described by the weak specification.

Input: A decision table $DT_1 = (U_1, A \cup A')$.

Output: CPN_{DS1} - the concurrent model of DS in the form of a CP-net.

Step 1: Extract the inhibitor rules from an underlying system S of DS and denote the set of such rules by $INH(S)$.

Step 2: Extract the inhibitor decision rules from a decision table DT_1 corresponding to dependencies of the form $\{a_1, ..., a_m\} \rightarrow \{a'_1\}, \ldots, \{a_1, ..., a_m\} \rightarrow \{a'_m\}$ and denote the set of such rules by $INH(DT_1)$.

Step 3: Construct the net representing all local processes in an underlying information system S. Each place p_a corresponds to a local process $a \in A$ of S (i.e. the number of places in the net is equal to the number of local processes in S). The colour sets of places in the net are labeled by means of the names of local processes (attributes) of S. For each place the colour set of place consists of colours labeled by means of the names of local states (attribute values) of a given process. There is only one transition t in the constructed net. The transition t of the net represents the global state changes (i.e. the next state relation). The initial marking M_0 of the net corresponds to the initial state of DS.

Step 4: The net obtained in Step 3 is extended by adding the guard expression to the transition t. Determine the guard expression from the inhibitor rules computed for an underlying system S and a decision table DT_1.

PROCEDURE 1 for computing a Boolean expression used to construct a guard expression.

Input: A set $INH(S)$ of inhibitor rules in S and a set $INH(DT_1)$ of inhibitor rules in DT_1.

Output: A Boolean expression G_{DS1} corresponding to the sets $INH(S)$ and $INH(DT_1)$.

Step 1: Rewrite each rule $x \Rightarrow \neg y$ from $INH(S)$, where x is the conjunction of descriptors on the left hand side of a rule and y is the conjunction of descriptors on the right hand side of a rule, using the Boolean algebra law $[x \Rightarrow \neg y] \Leftrightarrow [\neg(x \wedge y)]$, where $\Leftrightarrow$ denotes equivalence.

Step 2: Construct the conjunction of formulas obtained in Step 1.

Step 3: Use De Morgan law $[(\neg x_1) \wedge \ldots \wedge (\neg x_i)] \Leftrightarrow \neg[x_1 \vee \ldots \vee x_i]$ and the Boolean algebra law of the form $x \vee x \Leftrightarrow x$ (or dual) for simplification of the formula obtained in Step 2 and denote the Boolean expression corresponding to $INH(S)$ by G_S.

Step 4: Repeat Steps 1, 2, 3 for rules from $INH(DT_1)$ and denote the Boolean expression corresponding to $INH(DT_1)$ by G_{DT1}.

Step 5: Construct a Boolean expression G_{DS1} in the form: $G_{DS1} = G_S \wedge G_{DT1}$. G_{DS} is a Boolean expression corresponding to $INH(S)$ and $INH(DT_1)$.

The Boolean expression constructed in this way is used to construct the guard expression of a concurrent model. With each descriptor from G_S is connected an adequate variable of an output arc of the transition t. However for an expression G_{DT1}, with each descriptor corresponding to a previous state (i.e. in the form of $a_i = v_i$) is connected an adequate variable of an input arc of t and with each descriptor corresponding to a next state (i.e. in the form of $a'_i = v_i$) is connected an adequate variable of an output arc of t.

ALGORITHM 2 for constructing a concurrent model CPN_{DS2} of DS described by the strong specification.

Input: A decision table $DT_2 = (U_1, A_2 \cup \{d\})$.

Output: CPN_{DS2} - the concurrent model of DS in the form of a CP-net.

Step 1: Extract the inhibitor rules from an underlying system S of DS and denote the set of such rules by $INH(S)$.

Step 2: Extract the decision rules from a decision table DT_2 corresponding to dependency of the form $\{a_1, ..., a_m, a'_1, ..., a'_m\} \to \{d\}$. For each decision rule of the form $a_{i_1} = v_{i_1} \wedge \ldots \wedge a_{i_r} = v_{i_r} \wedge a'_{j_1} = v_{j_1} \wedge \ldots \wedge a'_{j_q} = v_{j_q} \Rightarrow d = 0$ (i.e. if $d = 0$) construct the inhibitor rule in the following form: $a_{i_1} = v_{i_1} \wedge \ldots \wedge a_{i_r} = v_{i_r} \Rightarrow \neg(a'_{j_1} = v_{j_1} \wedge \ldots \wedge a'_{j_q} = v_{j_q})$. Denote the set of such rules by $INH(DT_2)$.

Step 3: Construct the net representing all local processes in an underlying information system S like in Algorithm 1.

Step 4: The net obtained in Step 3 is extended by adding the guard expression to the transition t. Determine the guard expression from the inhibitor rules computed for an underlying system S and a decision table DT_2.

A procedure for computing a Boolean expression G_{DS2} used to construct a guard expression on the base of the set $INH(S)$ of inhibitor rules in S and the set $INH(DT_2)$ of inhibitor rules in DT_2 is formulated in an analogous way as a procedure presented above.

5 Extensions of dynamic information systems

If $S = (U, A)$ then a system $S' = (U', A')$ such that $U \subseteq U'$, $A' = \{a' : a \in A\}$, $a'(u) = a(u)$ for $u \in U$ and $V_a = V_{a'}$ for $a \in A$ will be called an *U'-extension* of S (or an *extension* of S, in short).

Let $S' = (U', A')$ be an U'-extension of $S = (U, A)$ and let $D(S)$ be a set of all rules in S. We say that S' is a *consistent extension* of S iff $D(S) \subseteq D(S')$.

S' is a *maximal consistent extension* of S iff S' is a consistent extension of S and for any consistent extension S'' of S' we have $D(S'') = D(S')$.

If $T \subseteq U \times E \times U$ then a transition relation $T' \subseteq U' \times E' \times U'$, where $U \subseteq U'$, $E \subseteq E'$ and $T'|_{U \times E \times U} = T$ will be called an *extension* of T.

Let $TS = (U, E, T, u_0)$ be a transition system. We say that a transition system $TS' = (U', E', T', u_0)$ is an *extension of the transition system* TS iff T' is an extension of T.

Let $DS = (U, A, E, T, u_0)$ be a dynamic information system. Let TS' be an extension of TS and let $D(TS)$ be a set of all decision rules in TS generated from a decision table DT_1 (or DT_2) described in Section 3. We say that TS' is a *consistent extension* of TS iff $D(TS) \subseteq D(TS')$, where $D(TS')$ is a set of all decision rules in TS'. TS' is a *maximal consistent extension* of TS iff TS' is a consistent extension of TS and for any consistent extension TS'' of TS' we have $D(TS'') = D(TS')$.

Let $DS = (U, A, E, T, u_0)$ be a dynamic information system, $S = (U, A)$ its underlying system and $TS = (U, E, T, u_0)$ its transition system. We say that a dynamic information system $DS' = (U', A, E', T', u_0)$ is an *extension of a dynamic information system* DS iff the following conditions are satisfied: $S' = (U', A)$ is an extension of S and $TS' = (U', E', T', u_0)$ is an extension of TS. DS' is a *maximal consistent extension* of DS iff S' is a maximal consistent extension of S and TS' is a maximal consistent extension of TS. A maximal consistent extension of a dynamic information system can be determined from its concurrent model CPN_{DS} and a concurrent model CPN_S of its underlying system S. Below we present a procedure for computing a maximal consistent extension DS' of DS on the base of concurrent models in the form of a *CP*-net.

PROCEDURE 2 for computing a maximal consistent extension DS' of DS:

Input: A dynamic information system $DS = (U, A, E, T, u_0)$ with its underlying system $S = (U, A)$.

Output: A maximal consistent extension DS' of the system DS.

Step 1: Construct the concurrent model CPN_S of the underlying system S in the form of a *CP*-net by using the method described in [5].

Step 2: Compute an extension $S' = (U', A)$ of the underlying system S of DS on the base of the reachability set of markings of CPN_S.

Step 3: Construct the concurrent model CPN_{DS} of the dynamic information system DS in the form of a *CP*-net by using Algorithm 1 or Algorithm 2.

Step 4: Compute the occurrence graph of CPN_{DS} (for example by using Design/CPN System [14]) with the initial marking corresponding to the initial state of DS.

Step 5: For each new global state from the extension S' for which there does not exist an adequate marking in occurrence graph computed in the previous step, compute the occurrence graph of CPN_{DS} with the initial marking corresponding to this state.

Step 6: Determine an extension T' of the transition relation T on the base of the occurrence graphs obtained in Step 4 and Step 5 in the follo wing way: For each occurrence graph, for each arc (M, b, M') of this occurrence graph, where $M[b > M'$ add adequate (u, e, u') to T' and e to E', where a global state u corresponds to the marking M and a global state u' corresponds to the marking M'. Do not take into consideration repeated arcs, i.e. arcs respected earlier in previous occurrence graphs.

Step 7: Construct the extension $DS' = (U', A, E', T', u_0)$.

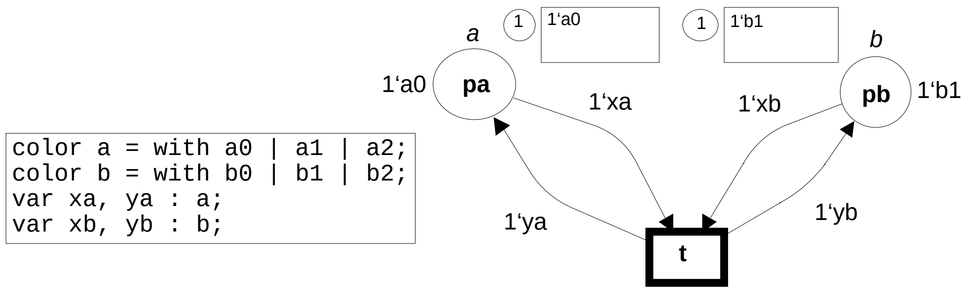
Example 5.1 Let us consider a dynamic information system DS and a decision table DT_1 from Example 3.1. The underlying system S of DS is presented in the first subtable of Table 1. By applying methods for generating the rules described in [12] for the system S we obtain the set of positive rules in the follo wingform: $OPT(S) = \{a_1 \Rightarrow b_0,\ a_2 \Rightarrow b_0,\ b_1 \Rightarrow a_0,\ b_2 \Rightarrow a_0\}$. On the base of this set we get the following set $INH(S)$ of inhibitor rules for this system: $INH(S) = \{a_1 \Rightarrow \neg b_1,\ a_1 \Rightarrow \neg b_2,\ a_2 \Rightarrow \neg b_1,\ a_2 \Rightarrow \neg b_2,\ b_1 \Rightarrow \neg a_1,\ b_1 \Rightarrow \neg a_2,\ b_2 \Rightarrow \neg a_1,\ b_2 \Rightarrow \neg a_2\}$. The Boolean expression G_S corresponding to this rules has the follo wingform: $G_S = \neg[(a_1 \wedge b_1) \vee (a_1 \wedge b_2) \vee (a_2 \wedge b_1) \vee (a_2 \wedge b_2) \vee (b_1 \wedge a_1) \vee (b_1 \wedge a_2) \vee (b_2 \wedge a_1) \vee (b_2 \wedge a_2)]$.

By applying the method of constructing concurrent models of information systems described in [5] we can construct the concurrent model CPN_S of this underlying system in the form of a CP-net and we can compute an occurrence graph of CPN_S b y using Design/CPN System [14]. The reachability set of markings of this net determines the maximal consistent extension S' of S. S' contains new object u_5 for which $a(u_5) = 0$ and $b(u_5) = 0$.

F or the decision tableDT_1 (see Table 2) we get the following set $INH(DT_1)$ of inhibitor decision rules: $b_1 \Rightarrow \neg a'_0,\ b_1 \Rightarrow \neg a'_1,\ a_1 \Rightarrow \neg a'_1,\ a_1 \Rightarrow \neg a'_2,\ b_2 \Rightarrow \neg a'_1,\ b_2 \Rightarrow \neg a'_2,\ a_2 \Rightarrow \neg a'_0,\ a_2 \Rightarrow \neg a'_2,\ b_1 \Rightarrow \neg b'_1,\ b_1 \Rightarrow \neg b'_2,\ a_1 \Rightarrow \neg b'_0,\ a_1 \Rightarrow \neg b'_1,\ b_2 \Rightarrow \neg b'_0,\ b_2 \Rightarrow \neg b'_2,\ a_2 \Rightarrow \neg b'_1,\ a_2 \Rightarrow \neg b'_2$.

After execution of the procedure 1 with the set $INH(DT_1)$ we obtain the follo wingBoolean expression: $G_{DT1} = \neg[(b_1 \wedge a'_0) \vee (b_1 \wedge a'_1) \vee (a_1 \wedge a'_1) \vee (a_1 \wedge a'_2) \vee (b_2 \wedge a'_1) \vee (b_2 \wedge a'_2) \vee (a_2 \wedge a'_0) \vee (a_2 \wedge a'_2) \vee (b_1 \wedge b'_1) \vee (b_1 \wedge b'_2) \vee (a_1 \wedge b'_0) \vee (a_1 \wedge b'_1) \vee (b_2 \wedge b'_0) \vee (b_2 \wedge b'_2) \vee (a_2 \wedge b'_1) \vee (a_2 \wedge b'_2)]$.

The concurrent model CPN_{DS1} of the dynamic information system DS in the form of a CP-net constructed b y using Algorithm 1 is shown in Figure 1. We must compute two occurrence graphs of CPN_{DS1} (see Figure 2 and 3). The first graph is fixed for an initial marking corresponding to the initial state of the system DS, whereas the second graph is fixed for an initial marking corresponding to the new state from an extension of the underlying system S. The marking corresponding to the new state is not reachable from the initial marking. In T able4 is shown a maximal consistent extension DS'_1 of DS specified by Table 2, obtained from occurrence graphs determined above.



[(not (((ya=a1) andalso (yb=b1)) orelse ((ya=a1) andalso (yb=b2)) orelse((ya=a2) andalso (yb=b1)) orelse ((ya=a2) andalso (yb=b2)))) andalso
(not (((xb=b1) andalso (ya=a0)) orelse ((xb=b1) andalso (ya=a1)) orelse ((xa=a1) andalso (ya=a1)) orelse ((xa=a1) andalso (ya=a2)) orelse ((xb=b2) andalso (ya=a1)) orelse ((xb=b2) andalso (ya=a2)) orelse ((xa=a2) andalso (ya=a0)) orelse ((xa=a2) andalso (ya=a2)) orelse ((xb=b1) andalso (yb=b1)) orelse ((xb=b1) andalso (yb=b2)) orelse ((xa=a1) andalso (yb=b0)) orelse ((xa=a1) andalso (yb=b1)) orelse ((xb=b2) andalso (yb=b0)) orelse ((xb=b2) andalso (yb=b2)) orelse ((xa=a2) andalso (yb=b1)) orelse ((xa=a2) andalso (yb=b2))))]

Fig. 1. The concurrent model CPN_{DS1} of DS in the form of a CP-net.

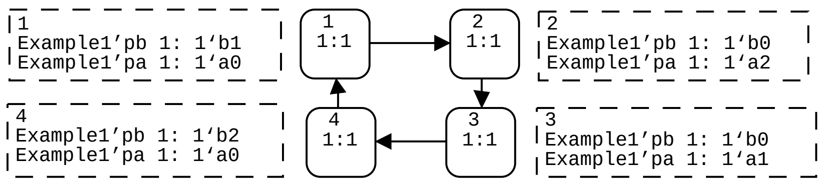


Fig. 2. The occurrence graph of CPN_{DS1} with an initial marking corresponding to the initial state.

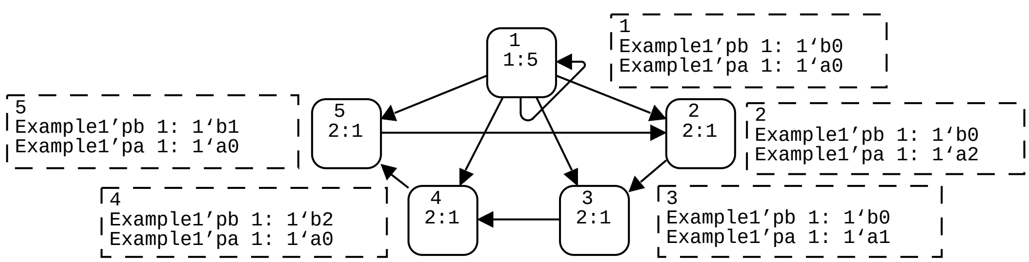


Fig. 3. The occurrence graph of CPN_{DS1} with an initial marking corresponding to the new state u_5.

Example 5.2 Let us consider a dynamic information system DS and a decision table DT_2 from Example 3.2. The maximal consistent extension of the underlying system S of DS has been computed in Example 5.1.

F or the decision tableDT_2 we get the following decision rules: $b_1 \wedge a'_0 \Rightarrow d_0$, $b_1 \wedge b'_1 \Rightarrow d_0$, $a_0 \wedge a'_1 \Rightarrow d_0$, $b_1 \wedge a'_1 \Rightarrow d_0$, $a_0 \wedge b'_2 \Rightarrow d_0$, $b_1 \wedge b'_2 \Rightarrow d_0$, $b_1 \wedge a'_2 \Rightarrow d_1$, $a_1 \wedge b'_1 \Rightarrow d_0$, $b_0 \wedge b'_1 \Rightarrow d_0$, $a_1 \wedge a'_1 \Rightarrow d_0$, $a_1 \wedge b'_0 \Rightarrow d_0$, $a_1 \wedge b'_2 \Rightarrow d_1$, $a_1 \wedge a'_2 \Rightarrow d_0$, $b_0 \wedge a'_2 \Rightarrow d_0$, $b_2 \wedge b'_1 \Rightarrow d_1$, $b_2 \wedge a'_1 \Rightarrow d_0$, $b_2 \wedge b'_0 \Rightarrow d_0$, $b_2 \wedge b'_2 \Rightarrow d_0$, $b_2 \wedge a'_2 \Rightarrow d_0$, $a_2 \wedge a'_0 \Rightarrow d_0$, $a_2 \wedge b'_1 \Rightarrow d_0$, $a_2 \wedge a'_1 \Rightarrow d_1$, $a_2 \wedge b'_2 \Rightarrow d_0$, $a_2 \wedge a'_2 \Rightarrow d_0$.

F or decision rules for which $d = 0$ we obtain the set $INH(DT_2)$ of inhibitor

Table 4
A maximal consistent extension DS'_1 of DS specified by Table 2

$U \backslash A$	a	b	$U \backslash E$	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
u_1	0	1		u_4								
u_2	1	0			u_3							
u_3	0	2				u_1						
u_4	2	0					u_2					
u_5	0	0						u_1	u_2	u_3	u_4	u_5

rules as follows: $b_1 \Rightarrow \neg a'_0$, $b_1 \Rightarrow \neg b'_1$, $a_0 \Rightarrow \neg a'_1$, $b_1 \Rightarrow \neg a'_1$, $a_0 \Rightarrow \neg b'_2$, $b_1 \Rightarrow \neg b'_2$, $a_1 \Rightarrow \neg b'_1$, $b_0 \Rightarrow \neg b'_1$, $a_1 \Rightarrow \neg a'_1$, $a_1 \Rightarrow \neg b'_0$, $a_1 \Rightarrow \neg a'_2$, $b_0 \Rightarrow \neg a'_2$, $b_2 \Rightarrow \neg a'_1$, $b_2 \Rightarrow \neg b'_0$, $b_2 \Rightarrow \neg b'_2$, $b_2 \Rightarrow \neg a'_2$, $a_2 \Rightarrow \neg a'_0$, $a_2 \Rightarrow \neg b'_1$, $a_2 \Rightarrow \neg b'_2$, $a_2 \Rightarrow \neg a'_2$.

On the base of the set $INH(DT_2)$ we create the following Boolean expression: $G_{DT2} = \neg[(b_1 \wedge a'_0) \vee (b_1 \wedge b'_1) \vee (a_0 \wedge a'_1) \vee (b_1 \wedge a'_1) \vee (a_0 \wedge b'_2) \vee (b_1 \wedge b'_2) \vee (a_1 \wedge b'_1) \vee (b_0 \wedge b'_1) \vee (a_1 \wedge a'_1) \vee (a_1 \wedge b'_0) \vee (a_1 \wedge a'_2) \vee (b_0 \wedge a'_2) \vee (b_2 \wedge a'_1) \vee (b_2 \wedge b'_0) \vee (b_2 \wedge b'_2) \vee (b_2 \wedge a'_2) \vee (a_2 \wedge a'_0) \vee (a_2 \wedge b'_1) \vee (a_2 \wedge b'_2) \vee (a_2 \wedge a'_2)]$.

The concurrent model CPN_{DS2} of a dynamic information system DS in the form of a CP-net constructed by using the Algorithm 2 is shown in Figure 4. We must compute two occurrence graphs of CPN_{DS2} like in Example 5.1. The first occurrence graph is the same as the graph in Figure 2 and the second occurrence graph is shown in Figure 5. In Table 5 is shown a maximal consistent extension DS'_2 of DS specified by Table 3, obtained from occurrence graphs determined above.

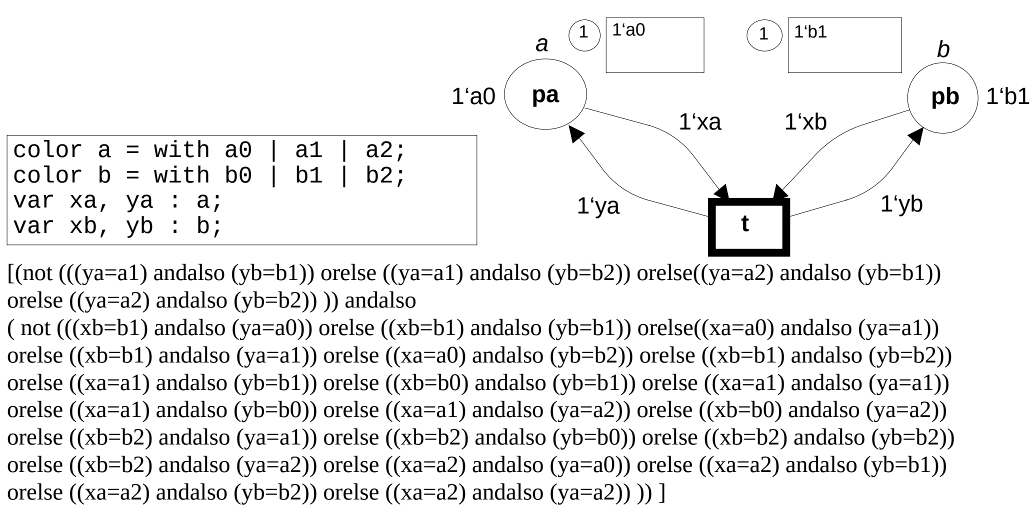


Fig. 4. The concurrent model CPN_{DS2} of DS in the form of a CP-net.

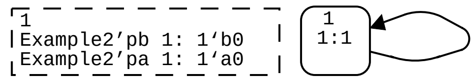


Fig. 5. The occurrence graph of CPN_{DS2} with an initial marking corresponding to the new state u_5.

Table 5
A maximal consistent extension DS'_2 of DS specified by Table 3

$U\backslash A$	a	b	$U\backslash E$	e_1	e_2	e_3	e_4	e_5
u_1	0	1		u_4				
u_2	1	0			u_3			
u_3	0	2				u_1		
u_4	2	0					u_2	
u_5	0	0						u_5

6 Conclusions

The paper presents two approaches to specifications and modelling of concurrent systems by means of dynamic information systems. The properties of concurrent models specified by a given dynamic information system depend on a chosen approach. The decision table of the second specification includes more information than the decision table of the first specification, thus the behaviour of such system is described more exactly. Hence an extension of a given dynamic information system for the second approach is smaller. The application of Coloured Petri Nets for modelling of concurrent systems specified by dynamic information systems enables us to discover in a simple way new states in which given dynamic information systems can find and to discover possible transitions between global states of given systems. The analysis of such models is simple by using Design/CPN system. The methods proposed in the paper have been implemented in the ROSECON system running on IBM PC computers under Windows operating system. The ROSECON system is being developed in the Chair of Computer Science Foundations at the University of Information Technology and Management in Rzeszow. In further investigations we will consider more exactly properties of presented specifications and obtained models of dynamic information systems and we will try to use other approaches to modelling of concurrent systems (for example cause-effect structures [1]).

References

[1] Czaja, L., "Elementary cause-effect structures," Warsaw University, 2002.

[2] Jensen, K., "Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol. 1," Springer. Berlin, 1992.

[3] Pancerz, K., and Z. Suraj, *R OSECON - asystem for automatic discovering of concurrent models from data tables.* In: Proceedings of the IX Environmental Conference on Mathematics and Informatics, Korytnica, June 12-16, 2002, Poland (in Polish), 34.

[4] Pancerz, K., and Z. Suraj, *From Data to Nets with Inhibitor Expressions: A R oughSet Approach.* In: Z. Suraj (ed.), Proceedings of the Sixth International Conference on Soft Computing and Distributed Processing, Rzeszow, June 24-25, 2002, P oland, 102-106.

[5] Pancerz, K., and Z. Suraj, *The Synthesis of Concurrent Systems Specified by Information Systems with Using the Coloured Petri Nets.* In: H.-D. Burkhard, L. Czaja, G. Lindemann, A. Sk owron and PStarke (eds.), Proceedings of the Workshop on Concurrency, Specification and Programming, Berlin, October 7-9, 2002, Germany, vol. 2, 255-266.

[6] Pawlak, Z., "Rough Sets - Theoretical Aspects of Reasoning About Data," Kluw er Academic Publishers. Dordrecht, 1991.

[7] Peters, J.F., A. Skowron, and Z. Suraj, *An Aplication of Rough Set Methods in Control Design,* Fundamenta Informaticae **43**(1-4) (2000), 269-290.

[8] Skowron, A., and Z. Suraj, *R ough Sets and Concurrency,* Bulletin of the Polish Academy of Sciences, **41**(3) (1993), 237-254.

[9] Skowron, A., and Z. Suraj, *Parallel Algorithm for R eal-Time Decision Making: A R oughSet Approach,* Journal of Intelligent Information Systems **7**, Kluwer Academic Publishers, Dordrecht (1996), 5-28.

[10] Suraj, Z., *Discovery of Concurrent Data Models from Experimental T ables: A R oughSet Approach,* Fundamenta Informaticae **28**(3-4) (1996), 353-376.

[11] Suraj, Z., *The Synthesis Pr oblemof Concurrent Systems Specified by Dynamic Information Systems,* In: L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery **2**, Physica-Verlag, Berlin (1998), 418-448.

[12] Suraj, Z., *R ough Set Methods for the Synthesis and Analysis of Concurrent Processes,* In: L. Polkowski, S. Tsumoto and T.Y. Lin (eds.), Rough Set Methods and Applications, Springer, Berlin (2000), 379-488.

[13] Suraj, Z., *Some R emarks on Extensions and R estrictions of Information Systems,* In: W. Ziarko and Y. Y ao (eds.), Rough Sets and Current T rends in Computing, Lecture Notes in Artificial Intelligence **2005**, Springer, Berlin (2001), 204-211.

[14] Design/CPN Web Page: `http://www.daimi.au.dk/designCPN/`.